Regular expressions
Регулярни изрази

ас. Стоян Мечев
катедра „Информационни технологии“
ВВМУ „Н. Й. Вапцаров“

Въведение

- Регулярният израз е специфичен шаблон, който описва набор от низове. Регулярните изрази се конструират аналогично на аритметичните изрази, като се използват различни оператори за комбиниране на по-малки изрази.
- Външно прилича на wildcard търсенето, но работи по-различно и предлага повече възможности.

Елементи на регулярните изрази

- Елементарни изрази от един символ
  - "a" отговаря на "a"
  - "abc" отговаря на "abc"
- Класове от символи - дефинира обхват от символи.
  - [a-z] един символ от всички малки букви от "a" до "z".
    - a[xy]b намира всички низове "axb" и "ayb" в текста.
  - [^abc] всички символи без a, b и c;
- Произволен символ ".". С точка е обозначава който и да е символ с изключение на символа за нов ред
  - Ако искате да укажете символа ".", използвайте "\. "
  - Помнете, че интервалът между думите също е символ
- Символ за начало на ред "^". Израз започващ с "^" означава "ред (абзац), който започва с израза, който следва след "^".
  - ^[КС] намира всички редове, които започват с "К" или "С".
- Край на ред "$". Израз завършващ с "$" означава "ред (абзац), който завършва с текста преди символа "$"
  - va$ намира всички редове, които завършват с "va"
- Повторение - знакът "*" означава, че изразът преди него може да се повтаря от 0 до неизвестен брой пъти.
  - аб* ще намери а, аб, абб, аббб и т.н.

Разширения на редовните изрази

- Думи
  - "\<" начало на дума (място, където небуквен знак среща буква)
  - "\>" край на дума (място където буква среща небуквен знак)
- Групиране "()". Изразът "а(бв)*" = а, абв, абвбв, абвбвбв и т.н.
- Алтернатива "|". Изразът (бял|царевичен|пълнозърнест) хляб = бял хляб, царевичен хляб, пълнозърнест хляб.
- Опционално "?" - изразът преди "?" се среща само 0 или 1 пъти. Изразът "а(бв)?" = само а или абв.
- Поне едно повторение "+" - също като "*", но трябва да има поне едно повторение. "а(бв)+" = абв, абвбв, абвбвбв и т.н., но не и само "а"
- Определен брой повторения "{n,m}". Ако е пропуснато n се приема 0, ако е пропуснато m се приема безкрайност.
  - ab{n} - точно n повторения на b;
  - ab{n,} - най-малко n повторения на b;
  - ab{,m} - най-много m повторения на b;
  - ab{n,m} между n и m повторения на b;
- Не-алчно съвпадение "*?", "+?", "??". Стандартните "*", "+" и "?" обикновено са "алчни", т.е. се опитват да намерят възможно най-много съвпадения, докато "не-алчните" оператори търсят възможно най-малко съвпадения.
  - алчно търсене "^a.*a" при входящ стринг "abacada", ще намери "abacada";
  - не-алчно търсене "^a.*?a" при входящ стринг "abacada", ще намери "aba";
- Не всички програми поддържат разширен синтаксис на регулярни изрази или има разлики в поведението им. Запознайте с документацията законкретната версия на съответната програма или програмен език!

Търсене във файлове и файлови системи

- grep (generalized regular expression processor)
- grep [OPTIONS] PATTERN [FILE...]
- **egrep ⇔grep -E** (--extended-regexp) приема шаблона като регулярен израз с разширен синтаксис;
  - **fgrep⇔grep -F** (--fixed-strings) приема шаблона като фиксиран низ;
- grep
  - Приема регулярен израз като задължителен параметър и списък с нула или повече файлове за търсене;
  - Ако не са дадени файлове, grep търси в stdin, което го прави филтър, който може да се използва в пайпинг;
  - Извежда на екран редовете, които отговарят на шаблона;
  - Ако няма съответстващи редове, grep не извежда съобщение, но генерира код за изход "0"
  - Има версии в които grep и egrep използват различни алгоритми за оценяване на регулярните изрази в резултат на което могат да имат доста различно бързодействие;
  - Най-добре е да слагате регулярния израз в единични кавички. Така ще избегнете риска от грешни резултати (обработка на израза като команда към шел).

Поддръжка на регулярни изрази

- Поддържа се;
  - Не се поддържа
    1. Изисква ескейп символ "\", напр. "аб\+" вместо "аб+"
    2. Няма нужда от скоби.
    3. Използва "\=" вместо "?".
    4. Съвсем друг синтаксис, вижте документацията.

Extension	GNU grep	GNU egrep	trad egrep	vim	emacs	Perl	Tcl
Word brackets	•	•	•	•1	•1	•4	•4
Grouping	•1	•	•	•1	•1	•	•
Alternative	•1	•	•	•2	•1	•	•
Option	•1	•	•	•3	•	•	•
At-least-once	•1	•	•	•1	•	•	•
Limits	•1	•	◦	•1	•1	•	•
Back-Reference	◦	•	•	◦	•	•	•
Non-Greedy	◦	◦	◦	•4	•	•	•

grep опции

-c (count) извежда само броя на съвпадащите редове;

-i (ignore) не прави разлика между главни и малки букви;

-l (list) извежда само имената на файловете, съдържащи текст, отговарящ на шаблона;

-n (number) номерира редовете, които извежда на екран;

-r (recursive) търси файлове и в поддиректориите;

-v (invert) извежда само редове, които не отговарят на шаблона;

-f (file) зарежда регулярния израз от файл. Ако файлът съдържа няколко реда, то всеки ред ще бъде обработен като отделен шаблон;

Примери

- Намерете всички думи съдържащи три пъти символа 'a'
  - grep '\<.*a.*a.*a.*\>' [filename]
- Намерете всички думи, започващи с 'a' и завършващи с 'a' или 'c'
  - grep '\<a.*[ac]\>' [filename]
- Намерете всички потребители на операционната система, чиито UID е с дължина 4 или 5 цифри
  - grep –E '\<[0-9]{4,5}\>' /etc/passwd
- Намерете всички имейли, с дължина точно 7 символа преди знака "@" и са от домейна ".com"
  - grep -E '^[a-z]{7}@.+\.com$' emails.txt
  - grep -E '^[a-z]{7}@.+\.com' emails.txt – грешен
  - grep -E '^[a-z0-9]{3,}@.*\.com$' emails.txt – грешен
- Какво ще изведат следните шаблони
  - grep -nE '\<[а-я]{4,}\>,
  - grep -E '\<[а-я]{,5}[0-9]{3}\>,
  - grep -E '\<[^абв][а-я]{,5}[0-9]{1}\>'

Благодаря за вниманието!

Въпроси?